IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA

Microsoft Corporation,

Plaintiff,

v.

Does 1-10 Operating an Azure Abuse Network,

Defendants.

Civil Action No.

FILED UNDER SEAL

DECLARATION OF RODELIO FIÑONES IN SUPPORT OF MICROSOFT'S MOTION FOR TEMPORARY RESTRAINING ORDER AND RELATED RELIEF

I, Rodelio Fiñones, declare as follows:

1. I am a Principal Security Software Engineer & Malware Researcher in Microsoft CELA Cybersecurity & Trust Engineering ("CSTE") and its former team Corporation's Digital Crimes Unit ("DCU"). My scope was expanded to also include AI and its abuse research. I make this declaration based upon my personal knowledge, and upon information and belief from my review of documents and evidence collected during Microsoft's investigation.

2. I have been employed by Microsoft since June 2009. In my role at Microsoft, I assess technological security threats to Microsoft and the impact of such threats on Microsoft's business and customers. I work with a team of investigators that focuses in part on researching different categories of malware, including botnets. My team and I research emerging malware threats through analysis of submitted samples, reverse engineering, forensic examination, data stream analysis, and development of tools to track botnet development. I am the team lead in developing malware prevention and eradication tools. Prior to joining Microsoft, I worked from 2004-2009 for Fortinet Technologies (Canada), Inc. as a Principal Software Developer/ Researcher (2007-2009) and Senior Antivirus Analyst (2004-2007). My job included research and analysis of complex malware and the development of tools to detect and eradicate malware.

From 1999-2004, I worked for Trend Micro, Inc. as a Senior Anti-Virus Researcher and Anti Virus Engine Developer. During my professional career, I have received advanced, specialized training and extensive practical experience in investigating malware and botnets and in devising technical countermeasures to detect and disable them. A true and current copy of my curriculum vitae can be found in Exhibit 2.

3. Since in or about August 2024, I have been part of the team investigating the attack on Microsoft's systems by the Defendants referred to in the Complaint in this case as the Azure Abuse Enterprise. My role to date has included observation, testing, and reverse engineering of the software Defendants used to carry out their attack on Microsoft's Azure OpenAI Service ("Attack"). My role has also included investigation into the Microsoft software and systems affected by the Attack in order to understand how Defendants gained access to and used the Azure OpenAI Services. My role has also included working with other investigators at Microsoft, including my colleagues Jason Lyons and Maurice Mason.

4. I regularly create, analyze, use, and reverse engineer software as part of my duties at Microsoft. Much of my work includes source code analysis. Source code refers to human-readable text for instructing a computer how to perform a function. When a software author sets out to write computer instructions in source code form, they have many options for how to express the instructions in human readable form. Factors that a software author may consider in determining how to express instructions in source code form include the ease with which another human can understand the code, the manner in which the code is organized, the way the code being authored relates to other portions of source code that it may interact with, security, and the ease of compiling the code into binary form (1's and 0's) for interpretation and execution by computers.

5. I am informed and believe based on my training and experience that software can be protected under the Copyright Act. I am informed and believe that copyright protection can extend to both source code and object code.[1] I am informed and believe that "[t]o qualify for

copyright protection, a work must be original to the author," meaning that "the work was independently created by the author (as opposed to copied from other works), and that it possesses at least some minimal degree of creativity."[2]

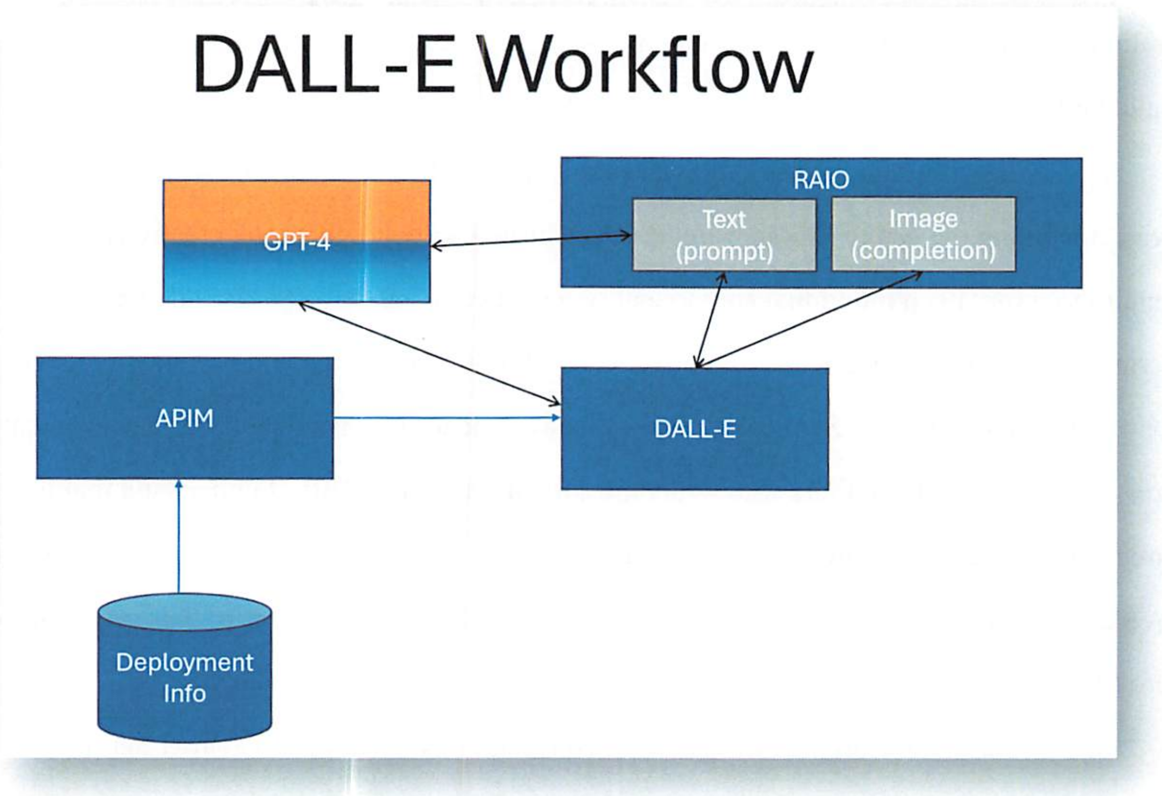
6. I have reviewed documentation concerning the software and hardware that operate the Azure OpenAI Service and have had discussions with developers and operators of the Azure OpenAI Service concerning the functionality, authorship, and ownership of Azure software accessed during Defendants attack, which I will generally refer to as the "Azure Middleware" that was relevant to my analysis. I have also reviewed how Defendants' de3u software and reverse proxy system interacted with Microsoft's systems to bypass the ordinary operation of certain Azure access and content controls and use the Azure OpenAI Service without authorization.

7. Based on my investigation and analysis, I understand that the Azure OpenAI Service, including the Azure Middleware, is sophisticated software that took creativity and ingenuity to write. Each individual component was created using the expertise and vast experience of Microsoft engineers, utilizing creative thinking to design and building a robust system. The Azure OpenAI Service as it exists today depends on many creative decisions that express choices made by authors who wrote the software for Microsoft. I understand that the Azure Middleware is software written for Microsoft that contains Microsoft copyright notices in source code header files, and that Microsoft considers itself to be the owner of the copyright to Azure Middleware.

8. Based on learnings from several conversations with Azure OpenAI Service engineers directly involved in planning, design, architecture, and coding of the Azure software discussed in this declaration, as well as learnings from my own investigations and reverse engineering of the actor toolsets described below and in the declaration of Jason Lyons, I have a strong technical understanding of the Azure OpenAI Service. The diagram below shows the overall architecture of the Azure OpenAI Service system responsible for the creation of images

using Microsoft's implementation of OpenAI's DALL-E model. Figure 1 below depicts this architecture.

Fig. 1: Azure OpenAI DALL-E workflow



DALL-E Workflow

9. To explain how the components of Figure 1 were accessed and used during the Attack, below I describe the steps of the communications process and what each component's role and function are during that process.

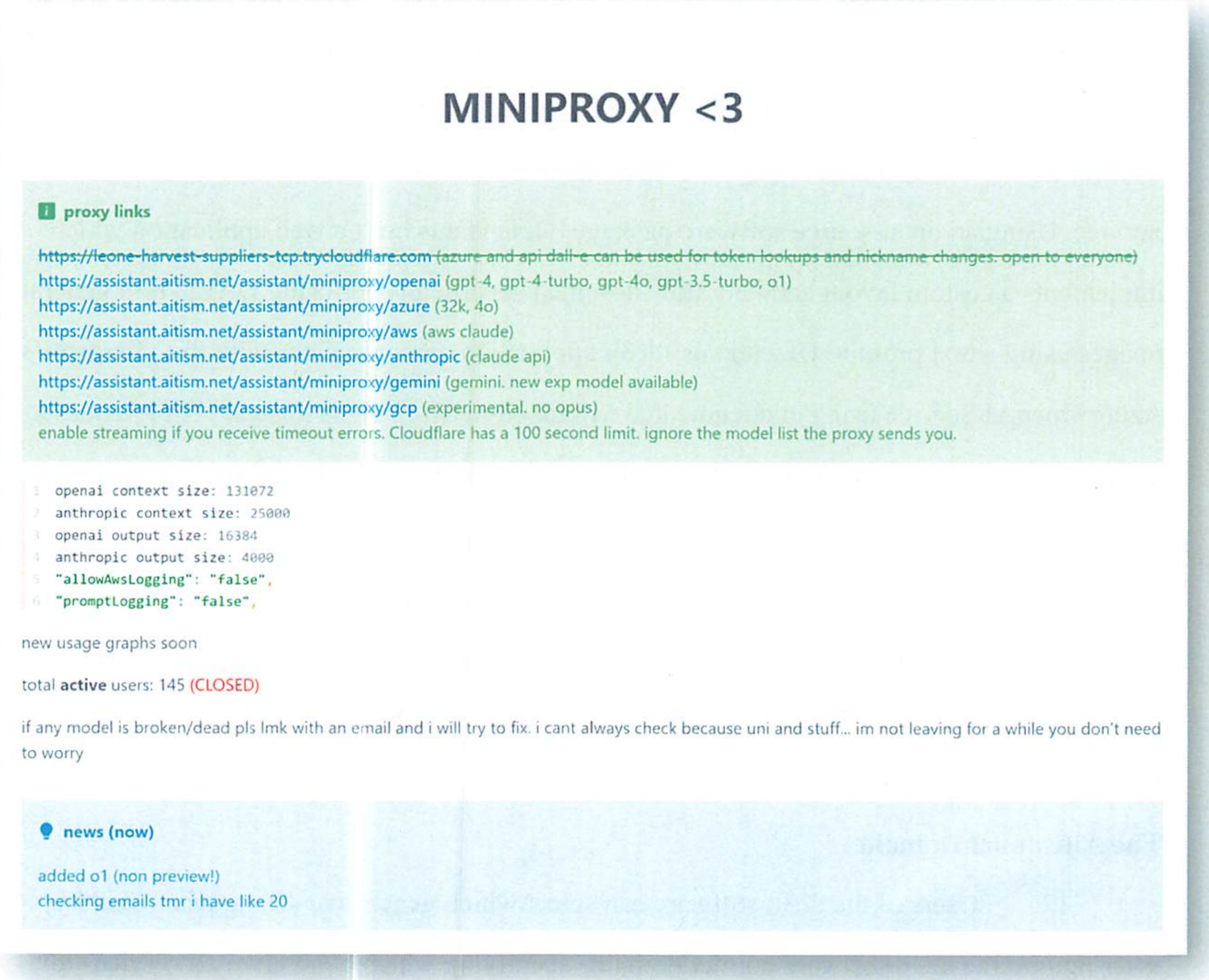
The de3u Client Software

10. The process starts with input by a user into the de3u software. De3u is custom software that allows users to issue API calls to generate images using the DALL-E model through a simple user interface that leverages the Azure APIs to access the Azure OpenAI Service. Using an open-source software package, Defendants built a web application which implements a custom layout and data flow designed specifically for using DALL-E to generate images using a text prompt. Defendants' de3u application communicates with the Microsoft's Azure OpenAI Service using undocumented Microsoft network APIs to send HTTP requests designed to mimic legitimate Azure OpenAI Service API requests. These requests are authenticated using stolen API keys and other authenticating information.

11. The de3u software converts user inputs into HTTP code that includes API calls, stolen API keys, and other authentication information. The de3u software sends the user-generated HTTP code to Defendants' reverse proxy tool, which alters the HTTP header and forwards the HTTP command to Azure OpenAI Service endpoints.

The Aitism.net Domain

12. Users of the de3u software can select which generative AI models to call by entering into the de3u software pointer domains specifying which generative AI services to communicate with. I understand that Microsoft's investigation resulted in identification of the pointer domains listed below on a website associated with Defendants infrastructure:



MINIPROXY <3

proxy links

~~https://leone-harvest-suppliers-tcp.trycloudflare.com (azure and api dall-e can be used for token lookups and nickname changes. open to everyone)~~
https://assistant.aitism.net/assistant/miniproxy/openai (gpt-4, gpt-4-turbo, gpt-4o, gpt-3.5-turbo, o1)
https://assistant.aitism.net/assistant/miniproxy/azure (32k, 4o)
https://assistant.aitism.net/assistant/miniproxy/aws (aws claude)
https://assistant.aitism.net/assistant/miniproxy/anthropic (claude api)
https://assistant.aitism.net/assistant/miniproxy/gemini (gemini. new exp model available)
https://assistant.aitism.net/assistant/miniproxy/gcp (experimental. no opus)
enable streaming if you receive timeout errors. Cloudflare has a 100 second limit. ignore the model list the proxy sends you.

```
openai context size: 131072
anthropic context size: 25000
openai output size: 16384
anthropic output size: 4000
"allowAwsLogging": "false",
"promptLogging": "false",
```

new usage graphs soon

total **active** users: 145 (CLOSED)

if any model is broken/dead pls lmk with an email and i will try to fix. i cant always check because uni and stuff... im not leaving for a while you don't need to worry

news (now)

added o1 (non preview!)
checking emails tmr i have like 20

13. When a user enters into de3u the pointer domain (URL) associated with the Azure OpenAI Service, the de3u software communicates that URL to oai reverse proxy service.

The "O-A-I" Reverse Proxy Service

14. The oai reverse proxy service includes custom designed software that facilitates communications between the de3u client software and Azure OpenAI Service. In addition to obfuscating the source of de3u client software communications, the oai reverse proxy service also reconfigures HTTP communications. For example, when a de3u user enters into the de3u

software the aitism.net pointer domain for the Azure OpenAI Service, the oai reverse proxy service parses the de3u HTTP communication, recognizes the Azure-related URL, and replaces the API key contained in the initial de3u HTTP communication with a stolen Azure API key and related authentication information before sending the revised HTTP communication to Microsoft's network.

Circumvention of Microsoft's Technological Access Control Measures

15. Defendants' de3u software permits users to circumvent technological controls that prevent alteration of certain Azure OpenAI Service API request parameters. For example, Microsoft's system is designed so that content generated using a given Microsoft customer's unique API key is only delivered to the endpoint address specified by that customer. Defendants' de3u software permits Defendants to send API calls configured by the reverse proxy service with rotating sets of stolen API keys and related authentication information in order to circumvent Microsoft's authentication gates.

16. Masquerading as a secured network HTTP request for low level API calls, the HTTP request is received from the oai reverse proxy service at the edge of Microsoft's systems and checked for authentication by the Azure "APIM" tool. This tool is run by Microsoft computers responsible for receiving, authenticating, and granting requests for API level access to the Azure OpenAI Service.

17. In the ordinary course of its operation, Microsoft's APIM system validates that the API request is coming from a source that is authorized and with sufficient privileges to perform the API call. Defendants malicious HTTP code and reverse proxy system allowed Defendants to use stolen API keys and altered endpoint information to bypass the normal operation of the APIM system and gain API access to the Azure middleware sitting behind the APIM authentication gate.

18. After getting through the APIM authentication system, Defendants' malicious HTTP request is passed by Azure middleware simultaneously to two groups of computers. One

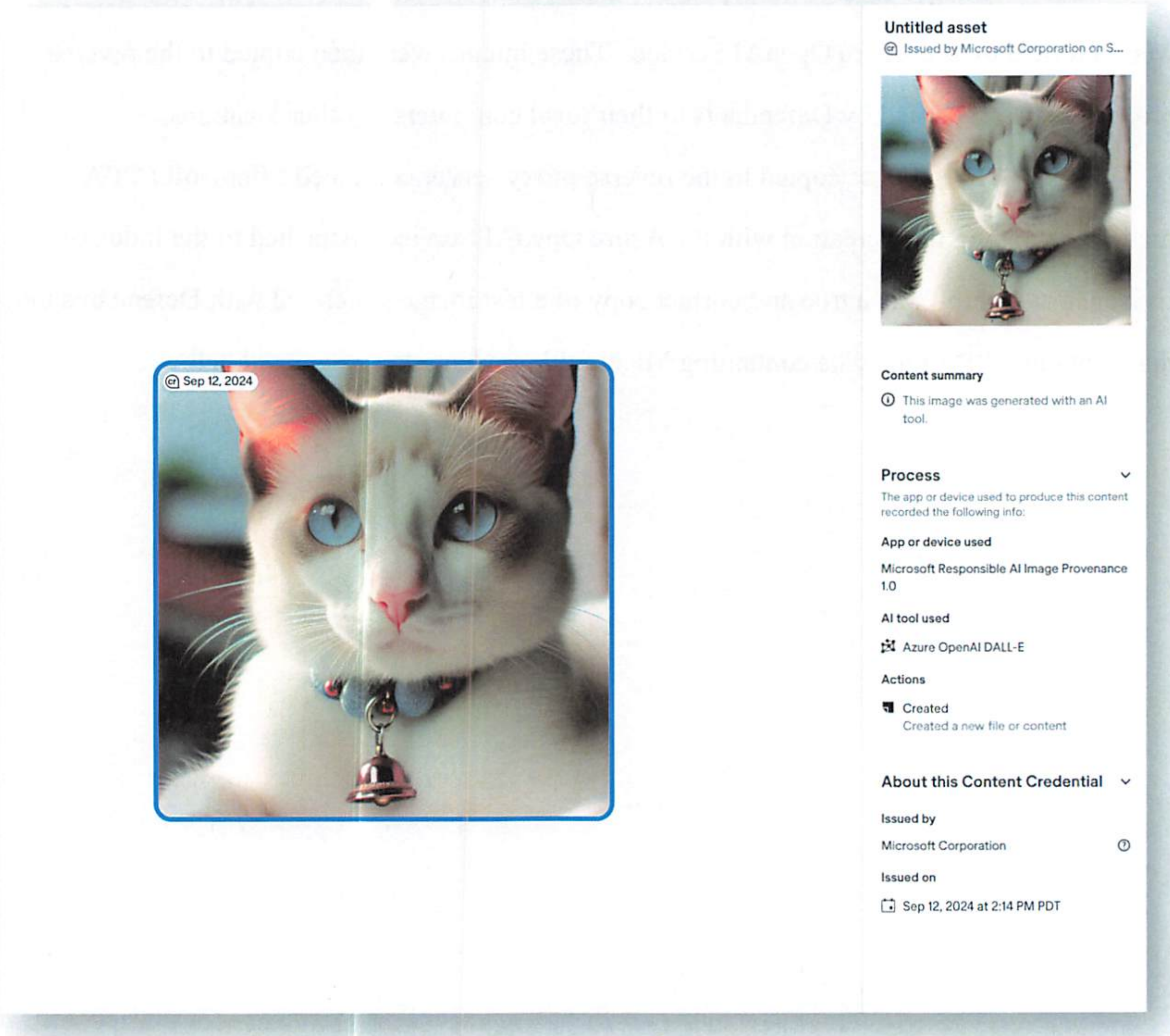
of these groups consists of the Microsoft owned computers that run Microsoft's licensed implementation of OpenAI's generative AI models like GPT and DALL-E ("Model Servers"). A DALL-E component processes the API request and calls GPT-4 to performs prompt transformation by analyzing the input prompt and generating a revised (re-write) prompt to pass to next stage of the process. The other group of computers consists of the computers running the Azure middleware software responsible for Microsoft's content and abuse filtering (I refer to this subclass of Azure middleware software and computers running the software as the "Azure Filtering Service"). The Model Servers and Azure middleware each simultaneously receive and process API calls contained in Defendants malicious HTTP requests, but ordinarily, responses to API calls are only returned after Azure Filtering Servers have implemented Microsoft's content and access controls.

19. The Azure OpenAI Service is designed so that in the ordinary course of its operation, all image generation prompts must be validated by default by the Azure Filtering Service before model outputs are returned in response to an API call. But Defendants' circumvented the Azure Filtering Service using stolen API keys and API access at scale, which allowed them to reverse engineer mechanisms for circumventing the Azure Filtering Service, for example, by using unicode characters, superscript, subscript, symbols, and other representations of letters ("encoding") that could not be fully parsed by the Azure Filtering Service. In this way, certain of Defendants' API calls that would have been rejected based on prompt text were encoded in a way that allowed them to bypass the Azure Filtering Service's validation checks. Without stolen API keys and API level access, Defendants could not have achieved these results.

20. Defendants' techniques and tools allowed them to access and use Microsoft computers and software that Defendants did not have authorization to access. Defendants then misused that access to create harmful images.

21. Images created in response to Defendants malicious HTTP code and API calls were returned by the Azure OpenAI Service. These images were then copied to the reverse proxy server and copied by Defendants to their local computers or other locations.

22. The images copied to the reverse proxy server contained Microsoft C2PA metadata showing their creation with the Azure OpenAI Service. Attached to the Index of Evidence as Exhibit 8 is a true and correct copy of a test image generated with Defendants tools that contains C2PA metadata containing Microsoft's trademark, as depicted below.



Untitled asset

Issued by Microsoft Corporation on S...

Content summary

This image was generated with an AI tool.

Process

The app or device used to produce this content recorded the following info:

App or device used

Microsoft Responsible AI Image Provenance 1.0

AI tool used

Azure OpenAI DALL-E

Actions

Created

Created a new file or content

About this Content Credential

Issued by

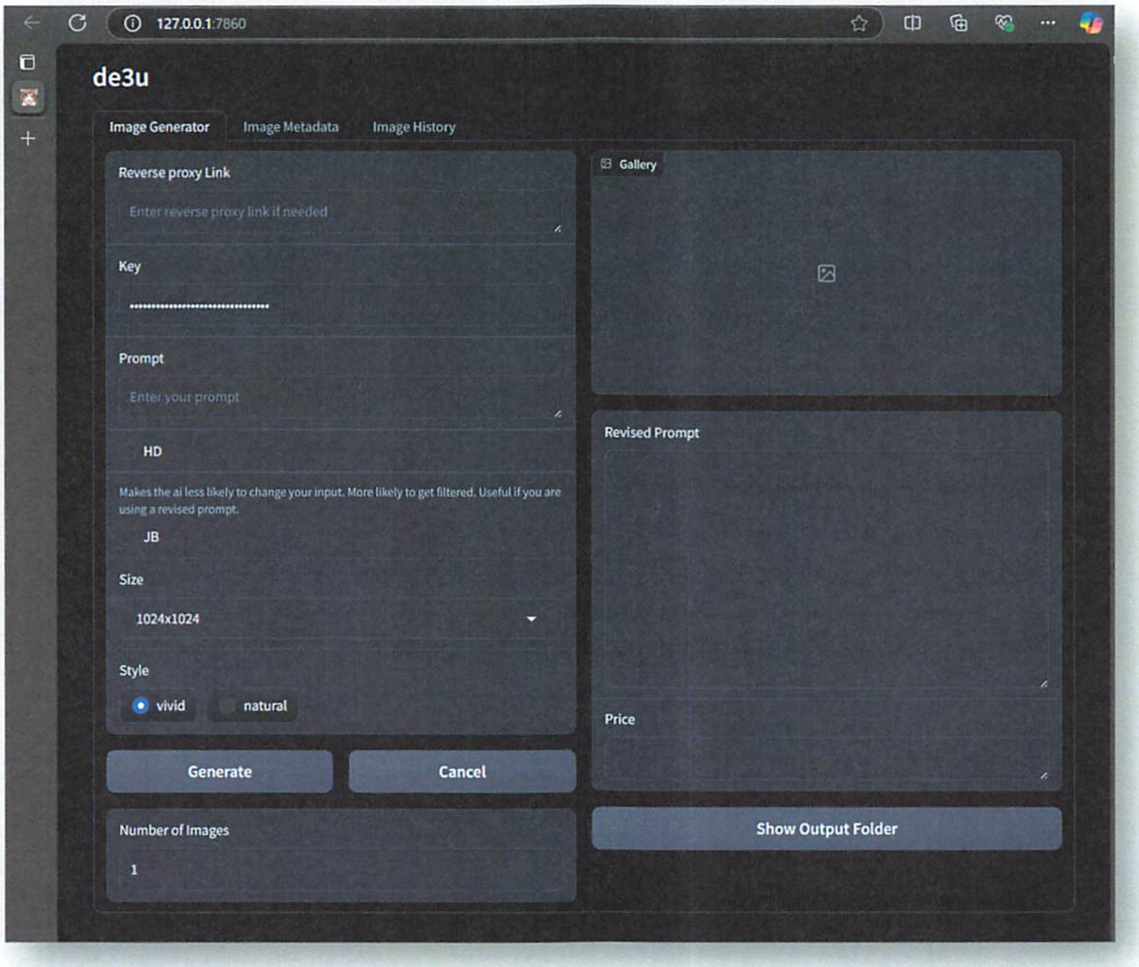
Microsoft Corporation

Issued on

Sep 12, 2024 at 2:14 PM PDT

23. However, my testing and analysis shows that de3u software can strip C2PA original image metadata, replacing it with custom metadata.

24. Attached to the Index of Evidence as Exhibit 9 is a true and correct copy of a screen shot of the de3u user interface, as depicted below.



Key: ••••••••••••••••••••••••••••••••

Makes the ai less likely to change your input. More likely to get filtered. Useful if you are using a revised prompt.

Size: 1024x1024

Style: ● vivid ○ natural

Number of Images: 1

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct to the best of my knowledge, information, and belief. Executed this 19th day of December, 2024 at Alexandria, Virginia.

Rodelio Fiñones